

S P R O C - A Multiple-Processor DSP IC

R. Davis
Hewlett-Packard ICBD
Corvallis, OR

Abstract- A large single-chip multiple-processor digital signal processing IC fabricated in HP-Cmos34 is presented. The innovative architecture is best suited for analog and real-time systems characterized by both parallel signal data flows and concurrent logic processing. The IC is supported by a powerful development system that transforms graphical signal flow graphs into production-ready systems in minutes. Automatic compiler partitioning of tasks among four on-chip processors gives the IC the signal processing power of several conventional DSP chips.

1 Introduction

Digital signal processing (DSP) involves the real-time acquisition of analog (continuous) inputs, their analysis and processing in a digital system, and subsequent synthesis and reintroduction back to the analog domain.

Conventional DSP chips are tuned for fast multiply and multiply-and-accumulate (MAC) algorithms on serial data streams such as required for filtering and spectral analysis. These algorithms take the ubiquitous form

$$y(n) = \sum_{i=1}^{N-1} a(i) * x(n-i) + \sum_{k=1}^M b(k) * y(n-k)$$

that compute outputs as weighted sums of present and past inputs, and past outputs. However, many analog and real-time systems are better characterized by complex networks of parallel, and often asynchronous, data flows and concurrent logic processing. Programming a conventional DSP chip to perform fundamental scheduling and synchronization tasks can become intractable.

SPROC¹, an IC and development system, efficiently manages concurrency through the use of dedicated control circuitry and a powerful compiler that automatically and transparently partitions tasks among several processors. It minimizes the number of components for simple systems, yet remains largely extensible for arbitrarily complex designs; it is easier to program with its library of customizable building blocks; it is easier to debug with its built-in real-time probe; it facilitates both rapid prototyping and production development on one system. It features full 24-bit fixed-point precision with 56-bit accumulation resulting in a 144dB dynamic range for signal bandwidths up to 250 kHz and handles all signal scaling automatically. The chip can be dynamically reprogrammed,

¹SPROC is the registered trademark of Star Semiconductor of Warren, NJ (908) 647-9400.

making adaptive, self-calibrating, and field upgradeable systems easier to design. The parallel port supports Motorola and Intel microprocessor interface protocols. The IC can be ganged to implement arbitrarily complex systems.

2 Chip Programming and Development Cycle

First, a signal-flow diagram of the desired system is graphically captured by selecting, placing, interconnecting, and parameterizing standard or customized function blocks, such as signal generators, summers, filters, etc. Next, the compiler converts the signal-flow diagram into executable code, allocating tasks efficiently between the available processors, building symbol tables for simple interfacing to the code. Then, the code is downloaded to the **SPROC** chip via either the development or target system. Finally, while the code is executing, circuit nodes can be probed, parameters can be modified, and the system observed in real time.

The **SPROC** advantages are fundamental: more complex, analog and real-time applications can be realized in a fraction of the time; designs can be observed in real-time and modified on-the-fly; any design that can be compiled is guaranteed to run on the **SPROC** chip. Higher designer productivity and improved performance translates into short time-to-market of more creative and competitive systems.

3 Chip Architecture

A *Harvard* architecture employing separate program and data busses allows concurrency in instruction fetch, decode, execution and data manipulation. The major blocks are the general signal processor (**GSP**), parallel interface (**HOST**), a serial interface (**ACCESS**), serial interfaces for sampled data (serial **PORTS**), a **DAC** port, a glue block (**GLUE**), and memory. An overview of the system architecture is shown in Figure 1.

SPROC operates in various configurations and modes. In *Master* mode, the system boots from external *EPROM*. In *Slave* mode, **SPROC** responds to an external controller which is either a microprocessor or a master **SPROC**. In *Redundancy* mode, the **GSPs** perform a system self-test, attempts redundancy and reconfigures the system. Thus, while the chip is highly integrated, it is flexible and extensible.

3.1 GSP

Each **GSP** is a 24-bit digital processor with 64 instructions and eight addressing modes. Main blocks include program control, address generator, multiplier, ALU, and decoder. Instructions include multiply (**MPY**) and multiply-and-accumulate (**MAC**) that execute in fifteen clock periods. One of up to four **GSPs** control both program and memory busses on a time-multiplexed basis. As triggered, a time slice for I/O operations via **HOST**, **ACCESS**, **PORTS**, or probing **DAC** is interjected. (see Figure 2)

P = Program Bus Access

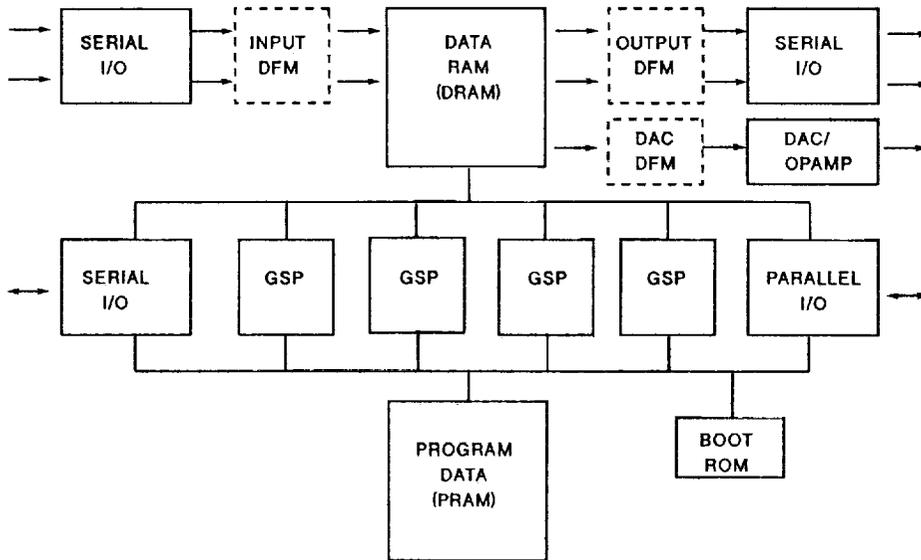


Figure 1: System Architecture

	cycle 0	cycle 1	cycle 2	cycle 3	cycle 4
GSP1	P		D		
GSP2		P		D	
GSP3			P		D
GSP4	D			P	
I/O		D			P

Figure 2: System Timing

3.3.4

D = Data Bus Access

I/O = HOST, ACCESS, PORTS, or probing DAC Access

In *Redundancy* mode, each **GSP** executes a self-test code from internal ROM upon power-up. If defective, the **GSP** is essentially held in reset and removed from tasking operations. This enables otherwise functional parts to yield at wafer test and provide fault-tolerance in the field. The fault coverage of this test is approximately 70%.

3.2 HOST

The host interface (**HOST**) is a 24-bit asynchronous bidirectional parallel port with a 64K addressing range, and supports 8, 16, and 24 bit transfers. It typically interfaces to the digital subsystem of the target environment. The **GSPs** can access the **HOST** via **LOAD** and **STORE** instructions. Internally, **SPROC** has a 12-bit addressing range with 4 bits reserved for master to slave addressing for memory-mapped devices or ganged **SPROCs**.

3.3 ACCESS

The access port (**ACCESS**) is a two port serial interface. It is typically used to observe and modify the contents of internal memory while the system is operating. The input port requires data, clock, and strobe; the output port drives a strobe and data based on the input port clock rate. Access is time multiplexed and is transparent to internal operations. Full read/write access is provided to any valid **SPROC** address.

3.4 PORTS

The sampled data streams are supported by four serial ports configurable for data, clock, strobe, and sync. There are two input and two output ports available. A data flow manager (**DFM**) manages the concurrency of multiple **GSP** and data RAM accesses. Very simply, an input **DFM** writes input sample data to consecutive data RAM locations and updates a write pointer. An output **DFM** will subsequently fetch output sample data from the data RAM.

3.5 GLUE

The glue block (**GLUE**) provides address decoding and memory mapping, mode control, system cycle generation, and serial port timing.

3.6 DAC

The digital-to-analog port (**DAC**) allows the probing of any node on the signal-flow diagram. These nodes are represented internally as two's complement FIFO buffers in data RAM. Hence, a node can be selected to direct its data buffer to the on-chip **DAC** port, and the analog value can be observed in real-time. An internal gain register can be loaded

to scale the digital value before outputting. The corresponding analog voltage is buffered and driven off chip, and may be observed with an oscilloscope, spectrum analyzer, etc.

4 IC Design Methodology

4.1 Partitioning

Star Semiconductor approached HP with a prototype system breadboarded with off-the-shelf memory and Xilinx and Actel field-programmable gate arrayed logic and a desire for fast, integrated silicon. Chip development on the customer side was primarily in *Cadence*; with VERILOG providing functional, behavioral, and logic simulation of the system and VERIFAULT for fault analysis. TA, a static timing analyzer was used for detailed timing optimization.

HP recommended developing additional standard cells including a recirculating flip-flop, adder, and lookahead cells to complement its standard cell offering HP-Cmos34. This resulted in enhanced performance, less silicon area, and a more direct mapping of the netlist. We also developed the memories, DAC, and OSC and the task of global composition and verification. Critical paths were simulated in SPICE, and capacitance was fed back to the customer for final timing simulations. Clock, power, and analog routing required manual editing.

4.2 New Standard Cell Development

Realizing the prevalent use of recirculating registers led to the incorporation of 2, 3, and 4-way multiplexers into the flip-flop to minimize area. (See table 1)

Table 1: Comparison of flip-flops, multiplexer combinations

	Library	Width uM	Intrinsic Delay nS	Load Multiplier nS/pF
DFFB	Standard	54.6	7.8	3.4
DFFF	Standard	121.8	2.6	1.5
X1RG1	New-Std	46.2	1.9	2.1
MUX2B	Standard	37.8	2.9	4.8
XMUX2	New-Std	33.6	1.8	1.3
X2RG1	New-Std	71.4	1.9	2.2

3.3.6

Also, adder cells were developed including a slow 1 bit adder for the multiplier, a fast 4 bit adder, and a 4 bit carry lookahead for the address logic. (See table 2)

Table 2: Adder cells

	Library	Width uM	Intrinsic Delay nS	Load Multiplier nS/pF
XADD1B	New-Std	63.8	4.2	2.4
XADD4	New-Std	226.8	1.8	3.6
XLOOK4H	New-Std	189.0	1.6	2.9

5 Composition

A standard methodology of composing chips with multiple standard cell and custom blocks with the autorouting (HARP) tools has been developed. First, blocks are routed with random port locations to determine size. Then, blocks are re-routed with assigned port locations determined by the floorplan. Finally, the top level is routed with the pads. Developing the SPROC chip produced some enhancements to the process.

5.1 Routing Tricks

Initial block sizes were estimated using the `csiz` program (which counts cells and adds their areas) with estimates for routing overhead. Port locations were assigned manually taking into account the initial floorplan and stored in a file for repeated runs and easy modification; random assignments were only made if a block had no assignment file. After iteratively routing to reach an optimal block size, a frame was extracted and placed in a dummy BDL file, which was then combined with custom frames for global routing including pads.

The new approach had the major advantage of flexibility of accepting new netlists from the designers and in experimenting with different partitions and floorplans in short order. Any piece could be easily rerouted and incorporated as desired, including the global route.

It was a must that each of the GSPs have optimal and identical performance, yet floorplan well. To accomplish this, ports were duplicated on each side of the block, and the blocks mirrored and routed back-to-back. To reduce the global routing, the block consisting of two GSPs only had one set of ports.

Routing ALLPORTS, INTERFACE, and GLUE as a single HARP block caused a great dispersal of the major busses. Partitioning these blocks and ports next to a central bussing channel proved to be more successful.

5.2 Routing Traps

Global power routing was problematic. Power estimates were determined by SPICE and the logic simulators. A package was selected to provide several power pads on each side. This required additional HARP modification. Also, end cap cells were modified to supply both power supplies to either end of the blocks, reducing IR drops by a factor of two. HARP was given parameters to increase the sizing of power busses between the blocks, each of which had multiple power ports. Manual editing was required to tie major power straps together, which run in pairs throughout the chip. The analog section was isolated by breaking the pad ring and connecting it to dedicated power pads. Also, digital signal lines were manually re-routed to avoid cross the analog logic.

Long global bussing of minimum width clock lines proved to have unaccepted RC wiring delays after final routing. The clock tree had to be resimulated taking these additional delays into account. To minimize skew, the clock drivers had been placed in the GLUE block, with the clock ports dispersed along one edge. The lines were selectively widened to a full contact width without penalty. It was sometimes possible to double the width of a single line if the vias on adjacent lines were coincident, or to drop the metal layers in parallel over long isolated runs. The clock network was reduced to a clock grid by effectively shorting the clock branches back together at the top level.

6 Custom Modules

6.1 RAM

The data and program memories are identical 1K word by 24-bit six-transistor static RAMs. A custom RAM was leveraged to improve the performance, as well as reduce area, with respect to an available RAM generator. The single-core array was developed for simplicity as 128 rows of 192 six-transistor static RAM columns. An 8-to-1 column multiplexer feeds a passive sense inverter and non-inverting tristate output buffer to achieve a 16ns cycle time in an area less than 10mm². About 80 % of the area is consumed by the core array. A dual clocking mode for precharge was adopted. In half-cycle mode, the timing is determined by two edges of the system clock up to 40MHz. In internal clock mode, an inverter delay chain times the precharge against one edge of a clock up to 50MHz. (4.75V, 85°C) With a 20ns cycle boundary, the address generation gate delays, wiring delays, and clock skew must be less than 4ns for 50MHz operation. Both RAMS are accessed every clock cycle and consume approximately 600mW each.

6.2 ROM

The internal ROM is 512 words by 24 bits. The core is organized as 64 rows and 192 columns. The cycle time for the ROM is less than 16ns. (4.75V, 85°C) The ROM address space overlaps the program RAM; while the system is booting the program RAM data drivers are disabled. The ROM artwork was logic simulated to verify the bit programming.

3.3.8

The ROM area is 0.84mm².

6.3 Analog Blocks

The OSC is an internal ring oscillator which minimizes component count for lower cost systems. The oscillator drives the system cycle generator when selected. An inverter feedback ring was chosen for simplicity. To reduce the frequency variability, the ring feedback is adjustable via programmable clocked-inverter taps decoded from three dedicated pins. The frequency variability is reduced to 36% over temperature and 17% over voltage over a tunable range of 30MHz to 80MHz. A schmitt trigger ring driver clocks a toggle flip-flop to insure a 50% duty cycle. In *Test* mode the oscillator is observable via a serial port. The oscillator resides in the pad ring to isolate it from the digital environment.

The DAC was selected from HP's customizable analog cell library available in HP-Cmos34. It is based on an 8-bit poly-resistor string design. Of note are Cmos transmissions gates used to make the resistor endpoints extendible to VDD and GND. The output swings between these voltage references which are sourced off-chip.

The OPAMP is a general purpose opamp that has a two-stage input and class AB output is used as a voltage follower to buffer the high-impedance DAC output. The opamp can swing rail-to-rail while driving a 3K resistive and/or 200pF capacitive load. An external compensation capacitor allows processing in Cmos34 without an extra mask required for linear capacitors.

7 Test Methodology

A 50MHz data rate speed goal made the Schlumberger S50 the local tester of choice. The customer contracted with TSSI (Beaverton,OR) for their software test development system (TDS) which converts captured simulation vectors to test vectors. TDS generates S50 MDC (patterns), TEG (timing), and pingroups directly. A pattern bridge (PBridge) essential samples the simulation responses, checking and formatting for S50 constraints. More than 900K vectors have been generated.

8 Results

First silicon was largely functional, with a major exception being the corruption of one of the processor addressing modes. Root cause was traced to a logic inversion in a Verilog model for a multiplexer. As a result, first silicon could not boot from ROM and hence run the redundancy code for self-test and configuration.

Second silicon was a quick, metal1/via/metal2 turn to correct the addressing mode, and the silicon was fully functional for software development and system operation up to 20MHz.

Third silicon was a full mask turn to increase the performance of the part. Unfortunately, a consequence of some of the edits introduced contention on the processor address

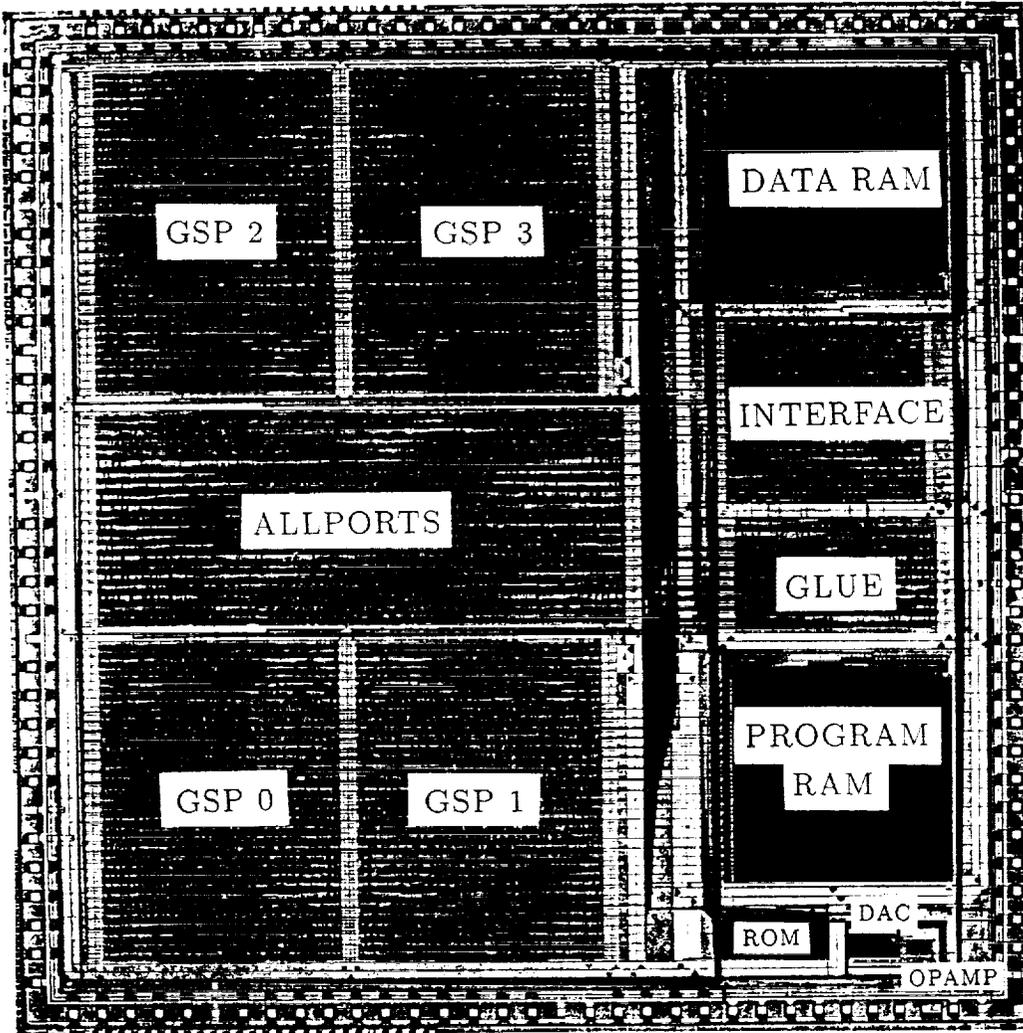
bus, limiting performance. Again, a quick turn is in the offing to solve the contention and improve the performance.

The 132 pin CPGA package can be fitted with a heatsink to allow operating the chip above 20MHz.

Investigation into porting the design into HP-Cmos26 are underway. The standard libraries are well-suited for 50MHz system operation, and the reduced silicon area will translate directly into a lower cost part and larger packaging offerings.

Conclusions

A large digital signal processing IC has been fabricated in HP-Cmos34. Routing processes have been improved, and the standard cell offering enhanced with additional cells. More accurate four-parameter timing models have been developed for Verilog and other industry simulators. New software was applied in the generation of a large set of test vectors. Sharing the design with the customer was largely successful without major show-stoppers resulting in beta-site quality systems on schedule. Efforts to port the design into HP-Cmos26 are underway promising higher performance and more competitive systems.



Die Size	13.7mm x 14.1mm
Routed Cells	56K gates
Custom RAM	48K bits
Custom ROM	12K bits
Total FETs	540,000
Package	600mil 132-CPGA
Power Supply	5.0V +/- 10%
Operating Power	2.5W (40MHz)

Table 3: Chip Characteristics and Photomicrograph